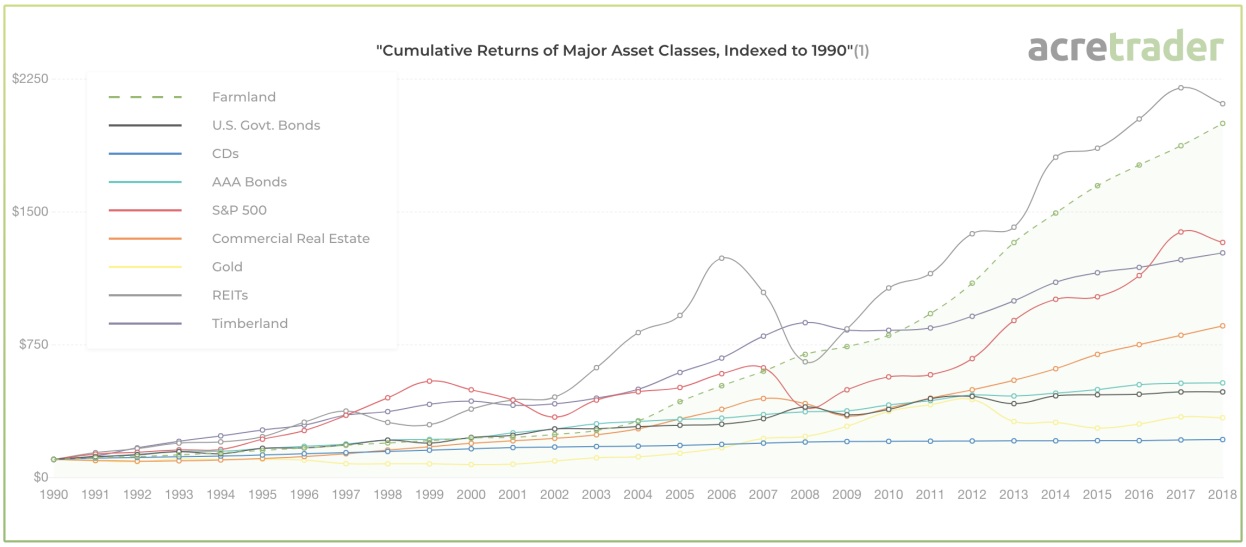
TEAM 3 - PROJECT 3

“MyFarm.org” powered by SegesCoin

Introduction

Land, and in particular farmland, is one of the oldest investment asset classes. It has been and continues to be an important source of wealth. Interestingly enough]gh, the largest farmland owner in the US today is Bill Gates with about 242,000 acres spread across 19 states.

Farmland investing requires very large resources and expertise, which is not easily accessible for the small or young investor.. There are some forms of land ownership opportunities via ETFs, REITSs and firms that sell participation in land portfolios but we believe that by using blockchain technology combined with machine learning techniques we can make access to farmland investing for the small and young investor not only possible but more cost efficient.



We also believe our investments not only will prove to be profitable but will also have a very strong positive social and environmental impact, which is a highly coveted investment feature particularly for investors coming from the Millenial and younger generations as many recent studies have shown.

Our farms and farmers will be required to join organizations like Leading Harvest (www.leadingharvest.org) and will be asked to maintain strict sustainable and organic agricultural standards, thus benefiting us all by protecting the environment while producing high quality foods. By schewing our investment towards small and family run farms, we also believe we'll have a strong positive social impact as well.

When you consider that more than 100 million Americans hail from the millennial and gen Z cohort alone that's a huge number of investors putting their money where their values are.

Project Description

We are creating a company called MyFarm (website www.MyFarm.org) that will allow small investors (targeting $5 to $25k initial investments) to participate in farmland investing. We will create a token (SegesCoin) using smart contracts in the ethereum network.

The token holder will benefit from potential price appreciation of the farmland holdings in the future, plus they may receive a pro rated percentage of the profits of each farm every year. Other potential source of income can be real estate brokerage fees by managing the land holdings.
Finally, we'll create a market for SegesCoin (SCN) also using smart contracts and list it in as many exchanges as possible to try to provide secondary market liquidity. SegesCoin (SCN) holders will be able to buy or sale SCN freely.

The management team will keep 6% fo the initial tokens offered (1% each) and charge an annual management fee of 1%,

Given the time constraint it is outside the scope of this project totoi study all the necessary regulatory and legal requirements to do all the above but it's clearly i\mportant to understand them.

Technology to be used

The following programs as required for Project 3 will be utilized:

- Blockchain (Remix, Ganache, Metamax, MyCoin)
- Machine learning (Multiple Python/Panda libraries)
- Mapbox plots

We'll also be utilizing other programs/websites from third party vendors:

- Speechelo (www.speechelo.com)

Workflow / Team assignments:

While every person in the team will be involved in all aspects of the project, we have divided the workflow as follow:

Deep	Smart Contracts design
Brendan/Alex/Philip	Machine learning
Elias	Video production/Smart Contracts design
Pablo	Google doc/Presentation/Idea coordination

Smart Contract Design

- Token issuance/creation(ERC20mintable)
- Registry & Verification
- Refundable
- Revenue distribution
- Market Creation

Machine learning

- Scrap API/Real Estate Websites for offers of farmland plots with the US that comply with the desired parameters (see below Land Acquisition Parameters)
- Plot land holdings and targeted land holdings on a map
- Create a land valuation model for buy/sell recommendations

Token Sale Parameters

- Crowdfunding aim to raise $20mm
- Min initial investment $5,000
- Max initial investment $25,000
- 2 year limit to invest/acquire land or money is returned (money raised to be kept in a separate account in 3m US Treasury bills if rates positive, cash otherwise)

Farmland Acquisition Parameters

- Option 1: buy and lease back
- Option 2: partial ownership (at least 25%) subject to production conditions
- Small plots (no more than 25 acres per plot)
- Closer to large urban centers
- Geographically diversify within the US
- Agriculturally diversified (25% livestock, 25% fish farms, 50% other)
- Production should comply with Leading Harvest standards (www.leadingharvest.org)
- Use of satellite technology from 3rd party vendors (https://www.beriqo.com)

```
pragma solidity ^0.5.0;

import "./SegesCoin.sol";
import "https://github.com/OpenZeppelin/openzeppelin-contracts/blob/release-v2.5.0/contracts/crowdsale/Crowdsale.sol";
import "https://github.com/OpenZeppelin/openzeppelin-contracts/blob/release-v2.5.0/contracts/crowdsale/emission/MintedCrowdsale.sol";
import "https://github.com/OpenZeppelin/openzeppelin-contracts/blob/release-v2.5.0/contracts/crowdsale/validation/CappedCrowdsale.sol";
import "https://github.com/OpenZeppelin/openzeppelin-contracts/blob/release-v2.5.0/contracts/crowdsale/validation/TimedCrowdsale.sol";
import "https://github.com/OpenZeppelin/openzeppelin-contracts/blob/release-v2.5.0/contracts/crowdsale/distribution/RefundablePostDeliveryCrowdsale.sol";

contract SegesCoinSale is Crowdsale, MintedCrowdsale, CappedCrowdsale, TimedCrowdsale,
RefundablePostDeliveryCrowdsale {

  constructor(uint rate, address payable wallet, SegesCoin token, uint open_time, uint close_time, uint goal, uint cap)

  Crowdsale (rate, wallet, token)

  TimedCrowdsale (open_time, close_time)

  CappedCrowdsale (cap)

  RefundableCrowdsale (goal)

  public
  {
    // constructor can stay empty
  }
}

contract SegesCoinSaleDeployer {

  address public token_sale_address;
  address public token_address;

  constructor(
    string memory name,
    string memory symbol,
    address payable wallet,
    uint goal
    )
    public
```

```
    {

      SegesCoin token = new SegesCoin (name, symbol, 0);

      token_address = address(token);

      SegesCoinSale seges_sale = new SegesCoinSale (1, wallet, token, now, now+24 weeks, goal,
300000000000000000000000);

      token_sale_address = address(seges_sale);

      token.addMinter(token_sale_address);
      token.renounceMinter();
  }
}
```

```
pragma solidity ^0.5.5;

import
"https://github.com/OpenZeppelin/openzeppelin-contracts/blob/release-v2.5.0/contracts/token/ERC20/ERC20.sol";
import
"https://github.com/OpenZeppelin/openzeppelin-contracts/blob/release-v2.5.0/contracts/token/ERC20/ERC20Detailed
.sol";
import
"https://github.com/OpenZeppelin/openzeppelin-contracts/blob/release-v2.5.0/contracts/token/ERC20/ERC20Mintabl
e.sol";

contract SegesCoin is ERC20, ERC20Detailed, ERC20Mintable {
    constructor(
        string memory name,
        string memory symbol,
        uint initial_supply
    )
        ERC20Detailed(name, symbol, 18)
        public
    {
        // constructor can stay empty
    }
}
```